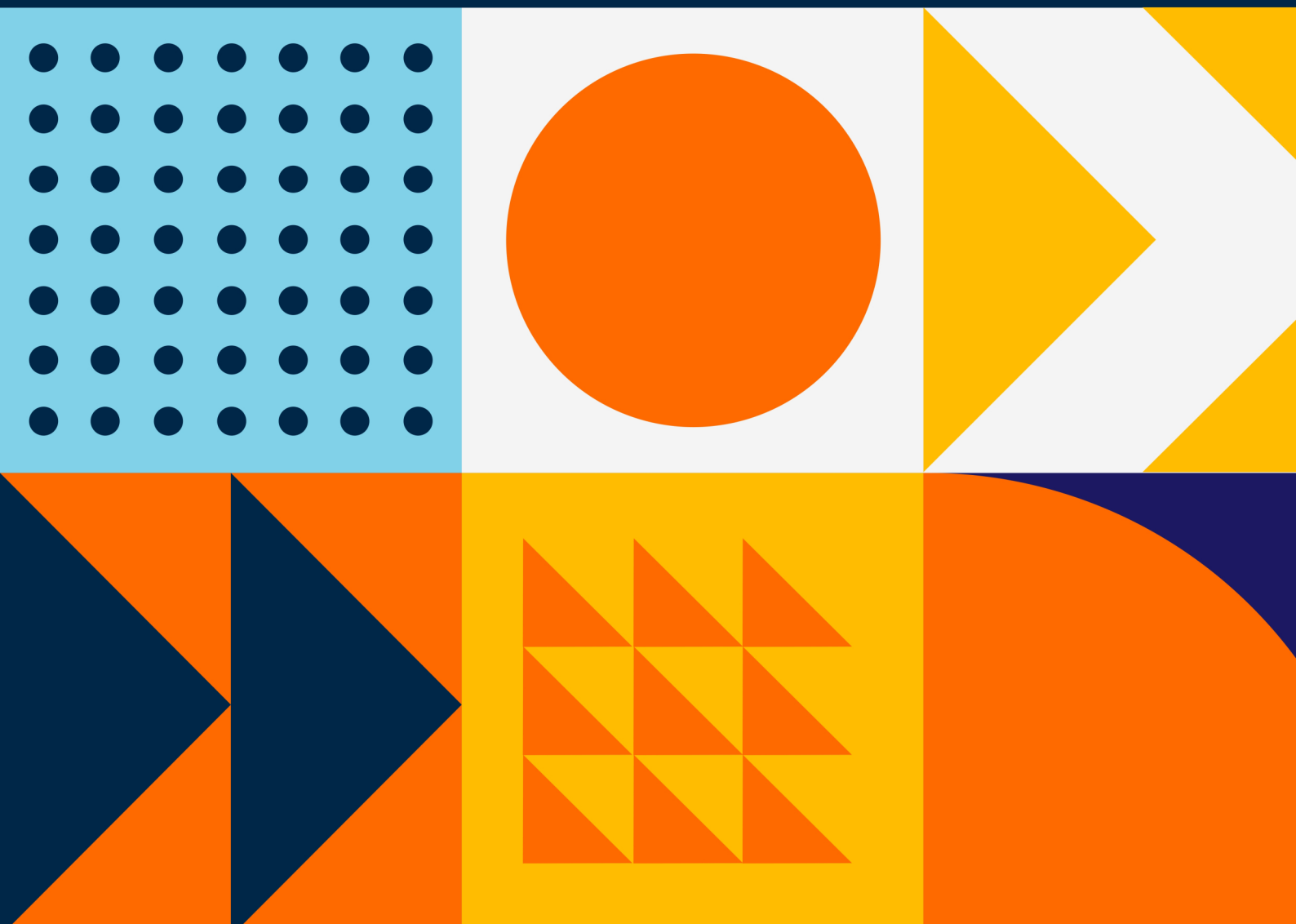
AVEREST
TRAINING & CONSULTING

We do IT
to inspire.

Agile, Scrum & DevOps:

Microsoft Azure DevOps Solutions



About Us

who we **are.**

Averest is one of the leading and fast-growing companies specialising in Information Technologies, Cyber Security, Cloud Computing, DevOps, Artificial Intelligence, Agile and Scrum, and Project Management, which is based in the United Kingdom and Turkey. Averest provides high-quality tech-accredited training and business solutions to its clients on these topics and more.

Why You Should Learn With Us?

We offer accredited Programs that are available for anyone wishing to acquire skills and gain professional certification to take their career to the next level.

- **100+ Premium Programs**

  Choose the appropriate program, date and region for your occupation.

- **50+ World-Wide Accredited Certifications**

  Get certified by global certification bodies and deepen your expertise.

- **500+ Expert Advisors**

  Get together with professional trainers who are experts in their professions.

- **100.000+ Professionals Trained**

  We help many of the world's leading companies to build their tech and digital capabilities.

Our partners.

ISACA

EC-Council

DevOps INSTITUTE

Project Management Institute

(ISC)²

IIBA®

OFFENSIVE security

CERTNEXUS

PECB

CompTIA

Microsoft

PeopleCert®
All talents, certified.

Scrum.org

aws

apm

ITIL®

To Explore More Please Visit **Our Website**

Overview

Microsoft Azure DevOps Solutions training course, delegates will learn how to design and implement DevOps practices for version control, compliance, and infrastructure as code. They will also learn configuration management, build, release, and testing using Azure technologies.

What You Will Learn?

- Describe the benefits of using source control
- Migrate from TFVC to Git
- Scale Git for Enterprise DevOps
- Implement and manage build infrastructure
- Manage application config & secrets
- Implement a mobile DevOps strategy
- Explain why continuous integration matters
- Implement continuous integration using Azure DevOps
- Configure builds and the options available
- Create an automated build workflow
- Integrate another build tooling with Azure DevOps
- Create hybrid build processes
- Please describe what is meant by code quality and how it is measured
- Detect code smells
- Integrate automated tests for code quality
- Report on code coverage during testing

Course Key Features

- Prepare for Microsoft MAZ-400 certification exam
- Microsoft Official Course content
- After-course instructor coaching benefit

Program

Microsoft Azure DevOps Solutions

- Add tooling to calculate technical debt
- Detect open source and other licensing issues
- Implement a container build strategy
- Differentiate between a release and a deployment
- Define the components of a release pipeline
- Explain things to consider when designing your release strategy
- Classify a release versus a release process, and outline how to control the quality of both
- Describe the principle of release gates and how to deal with release notes and documentation
- Explain deployment patterns, both in the traditional sense and in the modern sense
- Choose a release management tool
- Explain the terminology used in Azure DevOps and other Release Management Tooling
- Describe what a Build and Release task is, what it can do, and some available deployment tasks
- Classify an Agent, Agent Queue and Agent Pool
- Explain why you sometimes need multiple release jobs in one release pipeline
- Differentiate between multi-agent and multi-configuration release job
- Use release variables and stage variables in your release pipeline
- Deploy to an environment securely, using a service connection

Program

Microsoft Azure DevOps Solutions

- Embed testing in the pipeline
- List the different ways to inspect the health of your pipeline and release by using alerts, service hooks and reports.
- Create a release gate
- Describe deployment patterns
- Implement Blue-Green Deployment
- Implement Canary Release
- Implement Progressive Exposure Deployment
- Recommend artefact management tools and practices
- Abstract standard packages to enable sharing and reuse
- Inspect codebase to identify code dependencies that can convert to packages
- Identify and recommend standardized package types and versions across the solution
- Refactor existing build pipelines to implement version strategy that publishes packages
- Manage security and compliance
- Inspect open-source software packages for security and license compliance to align with corporate standards
- Configure build pipeline to access package security and license rating
- Configure secure access to package feeds
- Apply infrastructure and configuration as code principles

Program

Microsoft Azure DevOps Solutions

- Deploy and manage infrastructure using Microsoft automation technologies such as ARM templates, PowerShell, and Azure CLI
- Describe deployment models and services that are available with Azure
- Deploy and configure a Managed Kubernetes cluster
- Deploy and configure infrastructure using 3rd party tools and services with Azure, such as Chef, Puppet, Ansible, SaltStack, and Terraform
- Define an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructure
- Implement compliance and security in your application infrastructure
- Design practices to measure end-user satisfaction
- Design processes to capture and analyze user feedback from external sources
- Design routing for client application crash report data
- Recommend monitoring tools and technologies
- Recommend system and feature usage tracking tools
- Configure crash report integration for client applications
- Develop monitoring and status dashboards
- Implement routing for client application crash report data
- Implement tools to track system usage, feature usage, and flow
- Integrate and configure ticketing systems with the development team's work management system

Program

Microsoft Azure DevOps Solutions

- Analyze alerts to establish a baseline
- Analyze telemetry to establish a baseline
- Perform live site reviews and capture feedback for system outages
- Perform ongoing tuning to reduce meaningless or non-actionable alerts
- Plan for the transformation with shared goals and timelines.
- Select a project and identify project metrics and KPIs.
- Create a team and agile organizational structure.
- Develop a project quality strategy.
- Plan for secure development practices and compliance rules.
- Migrate and consolidate artefacts.
- Migrate and integrate source control measures.

Eligibility

Program

Microsoft Azure DevOps Solutions

Program Outline

Getting started with Source Control

- What is Source Control?
- Benefits of Source Control
- Types of source control systems
- Introduction to Azure Repos
- Migrating from TFVC to Git
- Authenticating your Git Repos

Scaling git for enterprise DevOps

- How to structure your git repo
- Git Branching workflows
- Collaborating with Pull Requests
- Why care about GitHooks?
- Fostering Internal Open Source
- Git Version
- Public projects
- Files in Git

Implement & Manage Build Infrastructure

- The concept of pipelines in DevOps
- Azure Pipelines
- Evaluate the use of Hosted vs Private Agents
- Agent pools
- Pipelines & Concurrency
- Azure DevOps and Open Source projects
- Azure Pipelines YAML vs Visual Designer

Managing application config & secrets

- Introduction to Security
- Implement a secure & compliant development process
- Rethinking application config data
- Manage secrets, tokens & certificates
- Implement tools for managing security and compliance in a pipeline

Program

Microsoft Azure DevOps Solutions

- Setup private agents
- Integrate Jenkins with Azure Pipelines
- Integration of external source control with Azure Pipelines
- Analyze & Integrate Docker multi-stage builds

Implement a mobile DevOps strategy

- Introduction to Mobile DevOps
- Introduction to Visual Studio App Center
- Manage mobile target device sets and distribution groups
- Manage target UI test device sets
- Provision of tester devices for deployment
- Create public and private distribution groups

Implementing Continuous Integration in an Azure DevOps Pipeline

- Continuous Integration Overview
- Implementing a Build Strategy
- Lab: Enabling Continuous Integration with Azure Pipelines
- Lab: Creating a Jenkins Build Job and Triggering CI

Module 7: Managing Code Quality and Security Policies

- Managing Code Quality
- Managing Security Policies
- Lab: Managing Technical Debt with Azure DevOps and SonarCloud
- Lab: Checking Vulnerabilities using WhiteSource Bolt and Azure DevOps

Implementing a Container Build Strategy

- Implementing a Container Build Strategy
- Lab: Existing .NET Applications with Azure and Docker Images

Program

Microsoft Azure DevOps Solutions

Design a Release Strategy

- Introduction to Continuous Delivery
- Release strategy recommendations
- Building a High-Quality Release pipeline
- Choosing a deployment pattern
- Selecting the right release management tool
- Lab: Building a release strategy
- -Differentiate between a release and a deployment
- -Define the components of a release pipeline
- -Explain things to consider when designing your release strategy
- -Classify a release versus a release process, and outline how to control the quality of both
- -Describe the principle of release gates and how to deal with release notes and documentation
- -Explain deployment patterns, both in the traditional sense and in the modern sense
- -Choose a release management tool

Set up a Release Management Workflow

- Create a Release Pipeline
- Provision and Configure Environments
- Manage And Modularize Tasks and Templates
- Integrate Secrets with the release pipeline
- Configure Automated Integration and Functional Test Automation
- Automate Inspection of Health
-
- Lab: Automating your infrastructure deployments in the Cloud with Terraform and Azure Pipelines
- Lab: Setting up secrets in the pipeline with Azure Key vault
- Lab: Setting up and Running Load Tests
- Lab: Setting up and Running Functional Tests
- Lab: Using Azure Monitor as release gate
- Lab: Creating a Release Dashboard
- -Explain the terminology used in Azure DevOps and other Release Management Tooling
- -Describe what a Build and Release task is, what it can do, and some available deployment tasks
- -Classify an Agent, Agent Queue and Agent Pool
- -Explain why you sometimes need multiple release jobs in one release pipeline
- -Differentiate between multi-agent and multi-

Program

Microsoft Azure DevOps Solutions

configuration release job

- -Use release variables and stage variables in your release pipeline
- -Deploy to an environment securely, using a service connection
- -Embed testing in the pipeline
- -List the different ways to inspect the health of your pipeline and release by using, alerts, service --hooks and reports
- -Create a release gate

Implement an appropriate deployment pattern

- Introduction to Deployment Patterns
- Implement Blue-Green Deployment
- Feature Toggles
- Canary Releases
- Dark Launching
- AB Testing
- Progressive Exposure Deployment
- Lab: Blue-Green Deployments
- Lab: Traffic Manager
- -Describe deployment patterns
- -Implement Blue Green Deployment

Designing a Dependency Management Strategy

- Introduction
- Packaging dependencies
- Package management
- Implement a versioning strategy
- Lab: Updating packages
- -Recommend artefact management tools and practices
- -Abstract standard packages to enable sharing and reuse
- -Inspect the codebase to identify code dependencies that can be converted to packages
- -Identify and recommend standardised package types and versions across the solution

Program

Microsoft Azure DevOps Solutions

- -Implement Canary Release
- -Implement Progressive Exposure Deployment
- -Refactor existing build pipelines to implement a version strategy that publishes packages
- -Manage security and compliance

Manage security and compliance

- Introduction
- Package security
- Open source software
- Integrating license and vulnerability scans
- Inspect open source software packages for security and license compliance to align with corporate standards
- Configure build pipeline to access package security and license rating
- Configure secure access to package feeds

Infrastructure and Configuration Azure Tools

- Learning Objectives
- Infrastructure as Code and Configuration Management
- Create Azure REsources using ARM Templates
- Create Azure Resources using Azure CLI
- Create Azure Resources by using Azure PowerShell
- Additional Automation Tools
- Version Control
- Lab Deploy to Azure using ARM templates
- Module Review Questions

Azure Deployment Models and Services

- Learning Objectives
- Deployment Models and Options
- Azure Infrastructure-as-a-Service (IaaS) Services
- Azure Automation with DevOps
- Desired State Configuration (DSC)

Create and Manage Kubernetes Service Infrastructure

- Learning Objectives
- Azure Kubernetes Service
- Lab Deploy and Scale AKS Cluster
- Module Review Questions

Program

Microsoft Azure DevOps Solutions

- Azure Platform-as-a-Service (PaaS) services
- Azure Service Fabric
- Lab Azure Automation - IaaS or PaaS deployment
- Module Review Questions

Third Party and Open Source Tools available with Azure

- Learning Objectives
- Chef
- Puppet
- Ansible
- Cloud-Init
- Terraform
- Lab Provision and configure an App in Azure Using X
- Module Review Questions

Implement Compliance and Security in your Infrastructure

- Security and Compliance Principles with DevOps
- Azure Security Center
- Lab Integrate a scanning extension or tool in an AZ DevOps pipeline/security centre
- Module Review Questions

Recommend and design system feedback mechanisms

- The inner loop
- Continuous Experimentation mindset
- Design practices to measure end-user satisfaction

Planning for DevOps

- Transformation Planning
- Project Selection
- Team Structures
- Lab: Agile Planning and Portfolio Management with Azure Boards

Program

Microsoft Azure DevOps Solutions

Planning for Quality and Security

- Planning a Quality Strategy
- Planning Secure Development
- Lab: Feature Flag Management with LaunchDarkly and AzureDevOps

Migrating and Consolidating Artifacts and Tools

- Migrating and Consolidating Artifacts
- Migrating and Integrating Source Control
- Lab: Integrating Azure Repos and Azure Pipelines with Eclipse
- -Design processes to capture and analyze user feedback
- -Design process to automate application analytics
- Lab: Integration between Azure DevOps and Teams
- Lab: Feature Flags
- -Design practices to measure end-user satisfaction
- -Design processes to capture and analyze user feedback from external sources
- -Design routing for client application crash report data
- -Recommend monitoring tools and technologies
- -Recommend system and feature usage tracking tools

Implement process for routing system feedback to development teams

- Implement tools to track system usage, feature usage, and flow
- Implement routing for mobile application crash report data
- Develop monitoring and status dashboards

Optimize feedback mechanisms

- Site Reliability Engineering
- Analyze telemetry to establish a baseline
- Perform ongoing tuning to reduce meaningless or non-actionable alerts
- Analyze alerts to establish a baseline

Program

Microsoft Azure DevOps Solutions

- Integrate and configure ticketing systems
- Blameless PostMortems and a Just Culture

Program

Microsoft Azure DevOps Solutions

Program Schedule

AVEREST